LISTING OF CLAIMS:

Claim 1. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

creating an overlay plane including the object as a function of the receiving step; and

displaying the object, in response to the request, by overlaying the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 2. (Previously Presented) The method according to claim 1, wherein the window is a markup language document.

Claim 3. (Previously Presented) The method according to claim 2, wherein the markup language document is an HTML document.

Claim 4. (Previously Presented) The method according to claim 2, wherein the markup language document is an XML document.

Claim 5. (Previously Presented) The method according to claim 1, wherein the software application is a Web browser.

Claim 6. (Previously Presented) The method according to claim 5, wherein the Web browser is at least one of Netscape Navigator, Netscape Communicator, and Microsoft Internet Explorer.

Claim 7. (Previously Presented) The method according to claim 1, wherein the receiving step includes receiving the request as a result of the user clicking on a hyperlink.

Claim 8. (Previously Presented) The method according to claim 1, wherein the receiving step includes receiving the request as a result of the user clicking on a banner.

Claim 9. (Previously Presented) The method according to claim 1, wherein the receiving step includes receiving the request as a result of the user clicking on a graphical icon.

Claim 10. (Previously Presented) The method according to claim 1, wherein the receiving step includes receiving the request as a result of the user initiating a click event.

Claim 11. (Previously Presented) The method according to claim 1, wherein the receiving step includes receiving the request as a result of the user initiating a rollover event.

Claim 12. (Previously Presented) The method according to claim 1, wherein the receiving step includes receiving the request as a result of the user initiating a timing event.

Claim 13. (Previously Presented) The method according to claim 1, wherein the receiving step includes receiving the request as a result of the user requesting a new window to be displayed.

Claim 14. (Previously Presented) The method according to claim 13, wherein the new window is defined by a markup language document.

Claim 15. (Previously Presented) The method according to claim 14, wherein the markup language document is an HTML document.

Claim 16. (Previously Presented) The method according to claim 14, wherein the markup language document is an XML document.

Claim 17. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

creating an overlay plane using at least one layer including the object as a function of the receiving step, wherein the layer is created using a layering functionality of the software application; and

displaying the object, in response to the request, by overlaying the layer in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 18. (Cancelled).

Claim 19. (Previously Presented) The method according to claim 17, wherein the layer is a DHTML layer.

Claim 20. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving a request for the object to be displayed in the window, the request being included in at least one of the definition of the window and a content for the window;

creating an overlay plane using at least one layer including a reference to the object as a function of the receiving step, wherein the layer is created using a layering functionality of the software application and the layer is hidden from a user; and

displaying the object, in response to the request, by overlaying the layer in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 21. (Previously Presented) The method according to claim 20, wherein the displaying step further comprises:

displaying the layer to the user in response to the request by overlaying the layer in the window, wherein the reference to the object initiates the

streaming of an object data to the layer and the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 22. (Previously Presented) The method according to claim 20, wherein the layer is a DHTML layer.

Claim 23. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving a request for the object to be displayed in the window, the request being included in at least one of the definition of the window and a content for the window;

creating an overlay plane using at least one layer including the object as a function of the receiving step, wherein the layer is created using a layering functionality of the software application and the layer is hidden from a user; and

displaying the object, in response to the request, by overlaying the layer in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 24. (Cancelled).

Claim 25. (Previously Presented) The method according to claim 23, wherein the layer is a DHTML layer.

Claim 26. (Previously Presented) The method according to claim 1, wherein the creating step further comprises:

creating an overlay image including the object as a function of the receiving step.

Claim 27. (Previously Presented) The method according to claim 26, wherein the displaying step further comprises:

displaying the object, in response to the request, by overlaying the created overlay image in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 28. (Previously Presented) The method according to claim 1, wherein the overlay plane utilizes semi-transparent edges.

Claim 29. (Previously Presented) The method according to claim 1, wherein the displaying step includes the step of:

using a transition effect to display the created overlay plane, wherein the transition effect is at least one of a transparent transition, a rotating object transition, a zoom transition, an animation transition, a wipe transition, a page curl transition, and a ripple transition.

Claim 30. (Previously Presented) The method according to claim 1, wherein the displaying step further comprises:

displaying the object, in response to the request, by overlaying the created overlay plane in the window, wherein the overlay plane is directly composited with the window without using functionality of the software application and wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 31. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving, by a plugin-control, a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

creating, by the plugin-control, an overlay plane including the object as a function of the receiving step; and

displaying the object in response to the request by overlaying, by the plugin-control, the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 32. (Previously Presented) The method according to claim 31, wherein the window is a markup language document.

Claim 33. (Previously Presented) The method according to claim 32, wherein the markup language document is an HTML document.

Claim 34. (Previously Presented) The method according to claim 32, wherein the markup language document is an XML document.

Claim 35. (Previously Presented) The method according to claim 31, wherein the software application is a Web browser.

Claim 36. (Previously Presented) The method according to claim 35, wherein the Web browser is at least one of Netscape Navigator, Netscape Communicator, and Microsoft Internet Explorer.

Claim 37. (Previously Presented) The method according to claim 31, wherein the receiving step includes receiving, by the plugin-control, the request as a result of the user clicking on a hyperlink.

Claim 38. (Previously Presented) The method according to claim 31, wherein the receiving step includes receiving, by the plugin-control, the request as a result of the user clicking on a banner.

Claim 39. (Previously Presented) The method according to claim 31, wherein the receiving step includes receiving, by the plugin-control, the request as a result of the user clicking on a graphical icon.

Claim 40. (Previously Presented) The method according to claim 31, wherein the receiving step includes receiving, by the plugin-control, the request as a result of the user initiating a click event.

Claim 41. (Previously Presented) The method according to claim 31, wherein the receiving step includes receiving, by the plugin-control, the request as a result of the user initiating a rollover event.

Claim 42. (Previously Presented) The method according to claim 31, wherein the receiving step includes receiving, by the plugin-control, the request as a result of the user initiating a timing event.

Claim 43. (Previously Presented) The method according to claim 31, wherein the receiving step includes receiving, by the plugin-control, the request as a result of the user requesting a new window to be displayed.

Claim 44. (Previously Presented) The method according to claim 43, wherein the new window is defined by a markup language document.

Claim 45. (Previously Presented) The method according to claim 44, wherein the markup language document is an HTML document.

Claim 46. (Previously Presented) The method according to claim 44, wherein the markup language document is an XML document.

Claim 47. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving, by a plugin-control, a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

creating, by the plugin-control, an overlay plane using at least one layer including the object as a function of the receiving step, wherein the layer is created using a layering functionality of the software application; and

displaying the object in response to the request by overlaying, by the plugin-control, the layer in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 48. (Previously Presented) The method according to claim 47, wherein the displaying step further comprises:

displaying the object in response to the request by overlaying, by the plugin-control, the layer in the window, wherein the layer is overlaid with the window using a software application provided mechanism for a display of layers and wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 49. (Previously Presented) The method according to claim 48, wherein the layer is a DHTML layer.

Claim 50. (Previously Presented) The method according to claim 47, wherein the displaying step further comprises:

displaying the object in response to the request by overlaying, by the plugin-control, the layer in the window, wherein the layer is overlaid in the window using a plugin-control provided mechanism for a display of content in the window bypassing a software application provided mechanism for a display of layers and wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 51. (Previously Presented) The method according to claim 50, wherein the layer is a DHTML layer.

Claim 52. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving, by a plugin-control, a request for the object, the request being initiated by a behavior of a user viewing the window;

creating, by the plugin-control, an overlay plane including the object as a function of the receiving step;

defining a layer using the software application provided functionality, wherein the layer definition is included in the definition of the window;

placing the created overlay plane in the defined layer; and

overlaying, by the plugin-control, the created overlay plane in the window.

Claim 53. (Previously Presented) The method according to claim 52, wherein the overlaying step further comprises:

overlaying the defined layer with the window, wherein the defined layer is overlaid with the window using the software application provided mechanism for the display of layers.

Claim 54. (Previously Presented) The method according to claim 53, wherein the layer is a DHTML layer.

Claim 55. (Previously Presented) The method according to claim 52, wherein the overlaying step further comprises:

overlaying the defined layer with the window, wherein the defined layer is overlaid with the window using a plugin-control provided mechanism for the display of the object in the layer with a window bypassing the software application provided mechanism for the display of layers.

Claim 56. (Previously Presented) The method according to claim 31, wherein the creating step further comprises:

creating, by the plugin-control, a layer wherein the layer is hidden from the user and the layer includes a reference to the object.

Claim 57. (Previously Presented) The method according to claim 56, wherein the displaying step further comprises:

displaying the layer to the user in response to the request by overlaying, by the plugin-control, the layer in the window, wherein the reference to the object initiates the streaming of an object data to the layer and wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 58. (Previously Presented) The method according to claim 57, wherein the layer is a DHTML layer.

Claim 59. (Previously Presented) The method according to claim 31, wherein the creating step further comprises:

creating, by the plugin-control, a layer wherein the layer is hidden from the user and the layer includes the object.

Claim 60. (Previously Presented) The method according to claim 59, wherein the displaying step further comprises:

displaying the layer to the user in response to the request by overlaying, by the plugin-control, the layer in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 61. (Previously Presented) The method according to claim 60, wherein the layer is a DHTML layer.

Claim 62. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving, by a plugin-control, a request for the object, the request being initiated by a behavior of a user viewing the window;

creating, by the plugin-control, an overlay plane including the object as a function of the receiving step;

defining a layer using the software application provided functionality, wherein the layer definition is included in the definition of the window, the layer is hidden from the user, and the layer includes a reference to the object;

placing the created overlay plane in the defined layer; and

overlaying, by the plugin-control, the created overlay plane in the window.

Claim 63. (Previously Presented) The method according to claim 62, wherein the overlaying step further comprises:

displaying the defined layer to the user, wherein the reference to the object initiates the streaming of the object data to the defined layer which is overlaid with the window using the software application provided mechanism for the display of layers.

Claim 64. (Previously Presented) The method according to claim 63, wherein the layer is a DHTML layer.

Claim 65. (Previously Presented) The method according to claim 62, wherein the overlaying step further comprises:

displaying the defined layer to the user, wherein the reference to the object initiates the streaming of the object data to the defined layer which is overlaid with the window using a plugin-control provided mechanism for the display of content with a window bypassing the software application provided mechanism for the display of layers.

Claim 66. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving, by a plugin-control, a request for the object, the request being initiated by a behavior of a user viewing the window;

creating, by the plugin-control, an overlay plane including the object as a function of the receiving step;

defining a layer using the software application provided functionality, wherein the layer definition is included in the definition of the window, the layer is hidden from the user, and the layer includes the object;

placing the created overlay plane in the defined layer; and

overlaying, by the plugin-control, the created overlay plane in the window.

Claim 67. (Previously Presented) The method according to claim 66, wherein the overlaying step further comprises:

displaying the defined layer to the user, wherein the defined layer is overlaid with the window using the software application provided mechanism for the display of layers.

Claim 68. (Previously Presented) The method according to claim 67, wherein the layer is a DHTML layer.

Claim 69. (Previously Presented) The method according to claim 66, wherein the overlaying step further comprises:

displaying the defined layer to the user, wherein the defined layer is overlaid with the window using a plugin-control provided mechanism for the display of content with a window bypassing the software application provided mechanism for the display of layers.

Claim 70. (Previously Presented) The method according to claim 31, wherein the creating step further comprises:

creating, by the plugin-control, a overlay image including the object as a function of the receiving step.

Claim 71. (Previously Presented) The method according to claim 70, wherein the displaying step further comprises:

displaying the object in response to the request by overlaying, by the plugin-control, the created overlay image in the window using the software application provided mechanism for the display of layers with the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 72. (Previously Presented) The method according to claim 70, wherein the displaying step further comprises:

displaying the object in response to the request by overlaying, by the plugin-control, the created overlay image in the window using a plugin-control provided mechanism for the display of content with the window bypassing the software application provided mechanism for the display of layers, wherein

the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 73. (Previously Presented) The method according to claim 31, wherein the overlay plane utilizes semi-transparent edges.

Claim 74. (Previously Presented) The method according to claim 31, wherein the displaying step includes the step of:

using a transition effect to display the created overlay plane, wherein the transition effect is at least one of a transparent transition, a rotating object transition, a zoom transition, an animation transition, a wipe transition, a page curl transition, and a ripple transition.

Claim 75. (Withdrawn) A method for overlaying an object in a window of a software application, comprising the steps of:

creating an overlay plane including the object;

overlaying the created overlay plane with the window;

displaying the window overlaid with the created overlay plane;

receiving a request, the request being initiated by a behavior of a user viewing the window overlaid with the created overlay plane; and

displaying the window without the overlay plane as a function of the receiving step.

Claim 76. (Withdrawn) A method for overlaying an object in a window of a software application, comprising the steps of:

creating, by the plugin-control, an overlay plane including the object;

overlaying, by the plugin-control, the created overlay plane with the window;

displaying, by the plugin-control, the window overlaid with the created overlay plane;

receiving, by a plugin-control, a request, the request being initiated by a behavior of a user viewing the window overlaid with the created overlay plane; and

displaying the window without the overlay plane as a function of the receiving step.

Claim 77. (Previously Presented) A system for overlaying an object in a window of a software application, comprising:

a program memory;

a storage device; and

a processor, wherein the processor is adapted to:

(i) receive a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

(ii) create an overlay plane including the object as a function of the receiving step; and

(iii) display the object, in response to the request, by overlaying the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 78. (Previously Presented) The system according to claim 77, further including:

a browser program and a browser plugin-control, executing in the processor, the browser plugin-control causing the processor to:

(i) receive a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

(ii) create an overlay plane including the object as a function of the receiving step; and

(iii) display the object, in response to the request, by overlaying the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 79. (Previously Presented) The system according to claim 78, wherein the browser program is at least one of Netscape Navigator, Netscape Communicator, and Microsoft Internet Explorer.

Claim 80. (Previously Presented) The system according to claim 78, wherein the browser plugin-control is defined using the Netscape Application Programming Interface (API).

Claim 81. (Previously Presented) The system according to claim 78, wherein the browser plugin-control is at least one of a Netscape Navigator plugin and a Netscape Communicator plugin.

Claim 82. (Previously Presented) The system according to claim 78, wherein the browser plugin-control is an ActiveX control.

Claim 83. (Previously Presented) A medium storing a set of instructions, the set of instructions capable of being executed by a processor to implement a method for overlaying an object in a window of a software application, the method comprising the steps of:

receiving a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

creating an overlay plane including the object as a function of the receiving step; and

displaying the object, in response to the request, by overlaying the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 84. (Previously Presented) A medium storing a set of instructions, the set of instructions capable of being executed by a processor to implement a method for overlaying an object in a window of a software application, the method comprising the steps of:

receiving, by a plugin-control, a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

creating, by the plugin-control, an overlay plane including the object as a function of the receiving step; and

displaying the object in response to the request by overlaying, by the plugin-control, the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device.

Claim 85. (Previously Presented) A method for initiating the overlaying of an object in a window of a software application, comprising the steps of:

clicking by a user an element of the window;

creating a layer as a function of the clicking step, wherein the layer includes the object; and

displaying the object by overlaying the created layer in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device and wherein the layer is directly composited in the window without using a layering feature of the software application.

Claim 86. (Previously Presented) The method according to claim 85, wherein the window is a Web page.

Claim 87. (Previously Presented) The method according to claim 85, wherein the software application is a Web browser.

Claim 88. (Previously Presented) The method according to claim 87, wherein the Web browser is at least one of Netscape Navigator, Netscape Communicator, and Microsoft Internet Explorer.

Claim 89. (Previously Presented) The method according to claim 85, wherein the clicking step includes clicking by a user a hyperlink in the window.

Claim 90. (Previously Presented) The method according to claim 85, wherein the clicking step includes clicking by a user a banner in the window.

Claim 91. (Previously Presented) The method according to claim 85, wherein the clicking step includes clicking by a user a graphical icon in the window.

Claim 92. (Previously Presented) The method according to claim 85, wherein the clicking step includes clicking by a user a hot spot on the window.

Claim 93. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

creating an overlay plane including the object as a function of the receiving step; and

displaying the object, in response to the request, by overlaying the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device and wherein the overlay plane is directly composited in the window without using a layering feature of the software application.

Claim 94. (Previously Presented) A medium storing a set of instructions, the set of instructions capable of being executed by a processor to implement a method for overlaying an object in a window of a software application, the method comprising the steps of:

receiving a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

creating an overlay plane including the object as a function of the receiving step; and

displaying the object, in response to the request, by overlaying the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device and wherein

the overlay plane is directly composited in the window without using a layering feature of the software application.

Claim 95. (Previously Presented) A method for overlaying an object in a window of a software application, comprising the steps of:

receiving, by a plugin-control, a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

creating, by the plugin-control, an overlay plane including the object as a function of the receiving step; and

displaying the object in response to the request by overlaying, by the plugin-control, the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device and wherein the overlay plane is directly composited in the window without using a layering feature of the software application.

Claim 96. (Previously Presented) A medium storing a set of instructions, the set of instructions capable of being executed by a processor to implement a method for overlaying an object in a window of a software application, the method comprising the steps of:

receiving, by a plugin-control, a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

creating, by the plugin-control, an overlay plane including the object as a function of the receiving step; and

displaying the object in response to the request by overlaying, by the plugin-control, the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device and wherein the overlay plane is directly composited in the window without using a layering feature of the software application.

Claim 97. (Previously Presented) A system for overlaying an object in a window of a software application, comprising:

a program memory;

a storage device; and

a processor, wherein the processor is adapted to:

(i) receive a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

(ii) create an overlay plane including the object as a function of the receiving step; and

(iii) display the object, in response to the request, by overlaying the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device and wherein the overlay plane is directly composited in the window without using a layering feature of the software application.

Claim 98. (Previously Presented) The system according to claim 97, further including:

a browser program and a browser plugin-control, executing in the processor, the browser plugin-control causing the processor to:

(i) receive a request for the object to be displayed in the window, the request being initiated by a behavior of a user viewing the window;

(ii) create an overlay plane including the object as a function of the receiving step; and

(iii) display the object, in response to the request, by overlaying the created overlay plane in the window, wherein the object is displayed in a manner that is independent of a movement of a pointing device and wherein the overlay plane is directly composited in the window without using a layering feature of the software application.